REMARKS

Claims 1-26 are pending in this application. Claims 16-22 were allowed in the Office Action dated May 26, 2004. Claims 2, 13 and 26 were objected to but were indicated to be allowable if rewritten in independent form. The remaining claims were rejected in the Office Action. Reconsideration of the rejected claims is requested in view of amendments made to the claims and in view of the arguments presented below.

Claim 1 was rejected under 35 USC 102(e) as being anticipated by Joy (US Pat. No. 6,341,347). Claim 1 is amended to include "the enabling including fetching an address vector associated with the second context." This feature does not appear to be shown by Joy. This feature was previously recited in dependent claim 2, which was indicated to be allowable if rewritten in independent form. It is submitted that claim 1, as amended, is allowable over Joy for similar reasons to those given for the allowability of claim 2.

Claims 2-10 depend from claim 1 and are therefore submitted to be allowable at least for depending from an allowable base claim.

Claim 11 was rejected under 35 USC 102(e) as being anticipated by Joy. Claim 11, as amended, includes the element "a set of shared registers associated with both the first pipeline and the second pipeline." This element does not appear to be shown by Joy. Joy teaches "designing pipeline registers (flops) with multiple storage bits. The individual bits of a flip-flop are allocated to a separate thread," column 10, lines 25-26. The example shown in Figure 4B shows storage elements that may be selected as alternatives depending on which thread is selected. "The switch is used to select a particular latch according to the thread identifier (TID) that is active," column 11, lines 53-55. Thus, these appear to be storage elements associated with particular threads and not shared storage elements. Figure 5 shows "'thread selectable' filp-flop substitution logic," column 13, line 6. "Each bit in the N-bit 'thread selectable' flip-flop substitution logic 500 corresponds to an individual machine state or a thread, forming what may be called a 'virtual CPU.' Only one bit is active at any time so that only one of a plurality of virtual CPUs occupy the pipeline at a time," column13, lines 55-59. Thus, there do not appear to be shared registers shown in either Figure 4 or Figure 5. Shared registers were not found elsewhere in Joy either. Therefore, claim 11 is submitted to be allowable over Joy.

Claims 12 - 15 depend from claim 11 and are therefore submitted to be allowable at least for depending from an allowable base claim.

Claim 23 was rejected under 35 USC 102(a) and (b) as anticipated by Golson (US Pat. No. 5,390,332). Claim 23 recites "a communication engine comprising a pipeline context switching microprocessor." Microprocessor 200 of Golson was cited as showing this feature because it is described as "multi-tasking and can execute multiple processes concurrently," column 7, lines 23-24. However, this does not appear to show microprocessor 200 of Golson to be a "pipeline context switching microprocessor" of claim 23. Golson explains the multi-tasking of microprocessor 200, "Host processor 100 interrupts the microprocessor 200 and causes the state of the current process (hereinafter referred to as Process 1) executing on microprocessor 200, to be saved and the state of the next process to be executed by the microprocessor 200 (hereinafter referred to as Process 2) to be restored in the registers of the microprocessor, such that the microprocessor 200 will continue to execute in the context of the second process," column 7, lines 61 - column 8, line 1. Golson does not appear to describe pipeline context switching in this portion. Certain prior art systems use a single pipeline and achieve concurrent tasks by reconfiguring the pipeline each time a new task is started. Switching between tasks may require flushing data for one task from registers and fetching data associated with another task from elsewhere. This may take many clock cycles. In such systems there is no pipeline context switching because there are no alternative pipelines to switch between. Golson appears to describe such a system. Therefore, it is submitted that this element of claim 23 is not shown in the prior art. Thus, claim 23 is submitted to be allowable.

Claims 6, 7 and 10 are rejected under 35 USC 103(a) as being unpatentable over Joy in view of Swoboda (US Pat. No. 6,553,513). The Office Action acknowledged "Joy did not specifically show flush of the instructions as claimed." Swoboda was cited for this feature. However, the motivation to combine these references is not understood. It appears that Joy teaches away from using flushing such as that taught by Swoboda. Joy appears to disclose a system that makes flushing unnecessary. Joy teaches, "The individual bits of a flip-flop are allocated to a separate thread. When a first thread stalls, typically due to a cache miss, the active bit of a flip-flop is removed from the pipeline pathway and another bit of the flip-flop becomes active. The states of the stalled thread are preserved in a temporarily inactive bit of the individual flip-flops in a pipeline stage." Column 10, lines 26-32. Thus, rather than flushing bits associated with one thread before loading bits for another thread, Joy appears to store bits for both threads in dedicated flip-flops and simply access the bits corresponding to the thread that is current. This keeps both sets of bits in an accessible location so that transfer and flushing of bits appears unnecessary. Flushing not only appears unnecessary according to the teaching of Joy, but it appears to be contrary to this teaching because flushing is an alternative technique used where bits associated with a thread must be loaded at a location for execution and must be flushed from that location before execution of the next thread. Because Joy teaches away from flushing and thereby teaches away from combination with Swoboda, it is submitted that claims 6-7 are allowable.

Claim 10 recites "executing a debug context without disturbing execution of the first or second pipelines." The Office Action cited Swoboda as showing a debug context. However, the claim element "without disturbing execution of the first or second pipelines" does not appear to be addressed in the Office action. The cited portion of Swoboda (column 14, lines 44-62) does not appear to disclose this limitation. The limitation has not been found elsewhere in Swoboda or in Joy either. Therefore, it is requested that this claim element be shown in the prior art or that the rejection be withdrawn.

Claim 23 was rejected under 35 USC 103(a) as being unpatentable over Golson in view of Bunnell (US Pat. No. 5,594,903). As discussed above, Golson does not appear to show a "communication engine comprising a pipeline context switching microprocessor." This feature has not been pointed out in Bunnell either. Bunnell appears to be cited for a different feature (that a SPARC workstation has a microprocessor). Because a pipeline context switching microprocessor is not shown by either reference, it is submitted that claim 23 is allowable over the combination of Golson and Bunnell.

Claims 24-26 depend from claim 23 and are therefore submitted to be allowable at least for depending from an allowable independent claim.

Claim 24 was rejected as unpatentable over Golson in view of Bunnell and further in view of Diepstraten (US Pat. No. 6,205,468). It was stated that it would have been obvious to use the single integrated circuit of Diepstraten for the circuit of Golson. However, Golson appears to teach away from having the identified system "formed as a single integrated circuit" as in claim

24. Golson relates to "the control of a microprocessor coupled to a host computer system as a peripheral device to provide emulation of a microprocessor system," column 1, lines 11-13. Thus, host processor 100 and processor 200 of Golson are physically separate units configured as host and peripheral devices. This arrangement of host and peripheral device appears contrary to having both devices formed on the same chip. Therefore, it is submitted that the combination of Golson and Diepstrataten is not suggested, but rather Golson teaches away from such a combination. Therefore, claim 24 is submitted to be allowable.

Claims 27 and 28 are added. Claims 27 and 28 depend from claims 8 and 1 respectively and are therefore submitted to be allowable at least for depending from an allowable base claim. Claims 27 and 28 concern methods for assigning priority to different pipelines. Claim 27 recites, "the context switch disable bit being dynamically configurable." Claim 28 recites, "the second pipeline status being dynamically configurable." Joy does not appear to show these features because Joy does not appear to show priority indicators that are dynamically configurable. Claims 27 and 28 are believed to be additionally allowable over Joy for this reason. Claims 27 and 28 are supported throughout the text and, in particular, at page 10, line 26 - page 11, line 31.

Accordingly, it is believed that this application is now in condition for allowance and an early indication of its allowance is solicited. However, if the Examiner has any further matters that need to be resolved, a telephone call to the undersigned attorney at 415-318-1163 would be appreciated.

Certificate of Mailing Under 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope address to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on August 26, 2004.

Franklin Dyer

Respectfully submitted,

8/26/04

Peter A. Gallagher Date
Reg. No.47,584